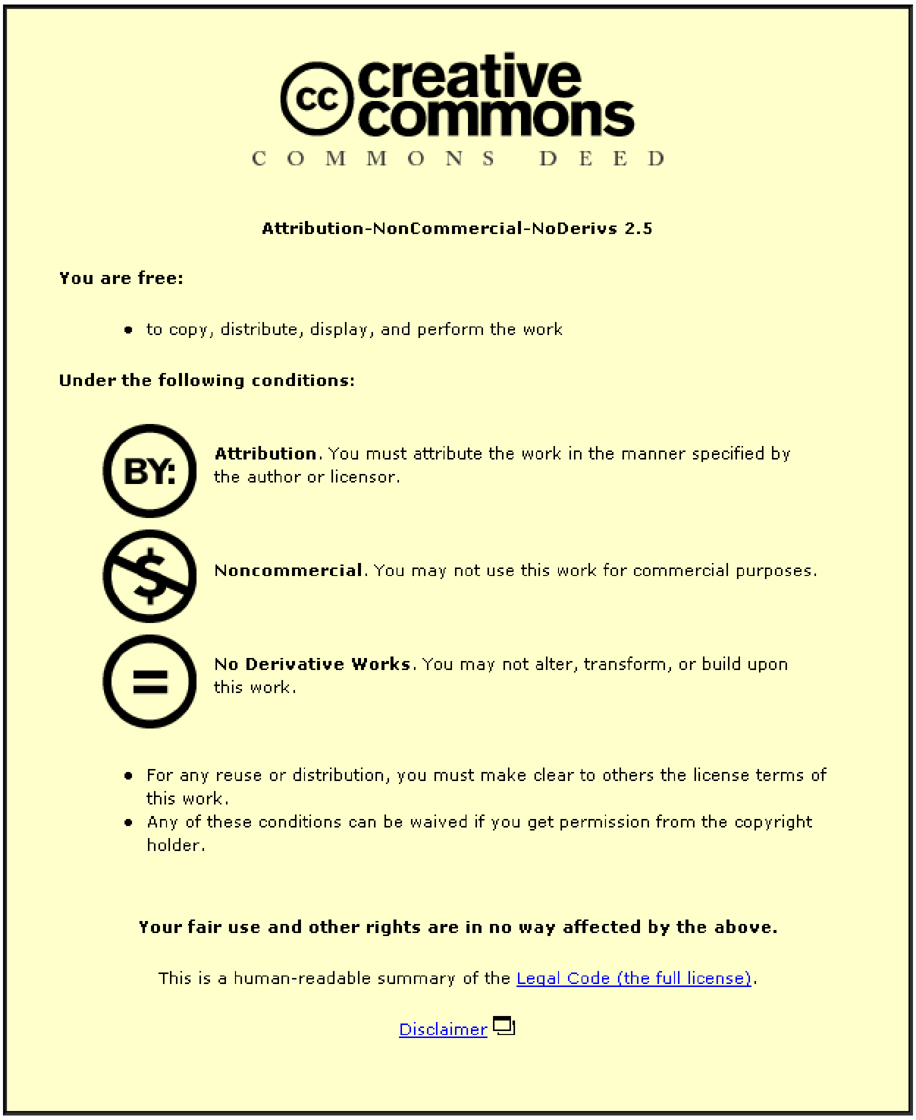

Inferring Descriptive Generalisations of Formal Languages

Dominik D. Freydenberger*
Goethe Universität,
Frankfurt am Main, Germany
freydenberger@em.uni-frankfurt.de

Daniel Reidenbach
Loughborough University,
Loughborough, United Kingdom
D.Reidenbach@lboro.ac.uk

Abstract

In the present paper, we introduce a variant of Gold-style learners that is not required to infer precise descriptions of the languages in a class, but that must find descriptive patterns, i. e., optimal generalisations within a class of pattern languages. Our first main result characterises those indexed families of recursive languages that can be inferred by such learners, and we demonstrate that this characterisation shows enlightening connections to Angluin's corresponding result for exact inference. Using a notion of descriptiveness that is restricted to the natural subclass of terminal-free E-pattern languages, we introduce a generic inference strategy, and our second main result characterises those classes of languages that can be generalised by this strategy. This characterisation demonstrates that there are major classes of languages that can be generalised in our model, but not be inferred by a normal Gold-style learner. Our corresponding technical considerations lead to deep insights of intrinsic interest into combinatorial and algorithmic properties of pattern languages.

1 Introduction

In Gold's intensively studied learning paradigm of language identification in the limit from positive data (Gold, 1967), it is a requirement for the computational learner to infer, for any positive presentation of any language in some class, an *exact* description of that language. While this maximum accuracy of the output of the inference procedure is clearly a natural goal, it has a number of downsides, the most obvious one being the fact that it can lead to significant limitations to the learning power of the model. From a more applied point of view, there is another important reason why one might wish to relax it and settle for receiving an approximation of the language from the learner: depending on the class of languages to be inferred, the corresponding grammars or acceptors might have undesirable properties, i. e., they might have computationally hard decision problems or be incomprehensible to a (human) user. Thus, in various settings it might be perfectly acceptable for an inference procedure to output a compact and reasonably precise approximation of the language instead of producing a precise yet arbitrarily complex grammar.

In the present paper, we introduce and study such a variant of Gold's model, where the requirement of exact language identification is dropped and replaced with that of inference of easily interpretable approximations. More precisely, we consider a learner that, for any language it reads, must converge to a *consistent pattern*, i. e., a finite string that consists of variables and of terminal symbols and that can be turned into any word of the language by substituting arbitrary strings of terminal symbols for the variables. In addition to being seen as mere descriptions of common features of words in a given language, such a pattern α can also be interpreted as a generator of a formal language $L(\alpha)$, the so-called *pattern language* (cf. Angluin (1980a)), which is simply the maximum set of words the pattern is consistent with. Hence, referring to this terminology, we can state that our learner has to output a pattern generating a language that is a superset of the input language, which means that our approach does not yield an arbitrary approximation of a language, but rather a *generalisation*. Even though many classes of pattern languages have a number of NP-complete or undecidable basic decision problems (see, e. g., Angluin (1980a), Jiang et al. (1994) and Freydenberger and Reidenbach (2010)), patterns (or related concepts, such as regular expressions and their various extensions implemented in today's programming languages and text editors, see Câmpeanu et al. (2003)) are widely used when commonalities of words are to be specified or interpreted by a human user, which demonstrates that they are a worthwhile concept in the context of our paper.

[*] Corresponding author.

When inferring consistent patterns instead of precise descriptions, it is of course vital to develop and employ a notion of high-quality patterns, so that the inference procedure does not lead to an overly imprecise result. Otherwise, the learner could always output the pattern $\alpha := x_1$ (where x_1 is a variable), which is consistent with every language, and this approach would obviously neither lead to a rich theory nor to practically relevant results. In our model, the inference procedure shall therefore be required to converge to a pattern δ that is *descriptive* of the language L (with respect to a class $\mathrm{PAT}_\star$ of pattern languages). This means that δ must be consistent with L, $L(\delta)$ must be included in $\mathrm{PAT}_\star$, and there is no pattern δ' satisfying $L(\delta') \in \mathrm{PAT}_\star$ and $L \subseteq L(\delta') \subset L(\delta)$; in other words, a pattern is descriptive of a language if there is no other pattern providing a closer match for the language. Since descriptiveness captures a natural understanding of patterns providing a desirable generalisation of languages and, furthermore, descriptive patterns can be used to devise Gold-style learners precisely identifying classes of pattern languages from positive data, this concept has been thoroughly investigated (see, e. g., Angluin (1980a), Jiang et al. (1994) and Freydenberger and Reidenbach (2009)). While established definitions of descriptiveness often restrict their view to patterns covering finite languages and normally use the full class of E- or NE-pattern languages (to be formally introduced in Section 2) as the class $\mathrm{PAT}_\star$ of admissible pattern languages, we allow a descriptive pattern to cover a finite or an infinite language, and we have a class $\mathrm{PAT}_\star$ that can be arbitrarily chosen. Both of these extensions of the original definition are absolutely straightforward.

To summarise our model of inference, we consider a learner that reads a positive presentation of a language and, after having seen a new input word, outputs a pattern, the so-called *hypothesis*. We then say that, for a class $\mathcal{L}$ of languages and a class $\mathrm{PAT}_\star$ of pattern languages, the learner $\mathrm{PAT}_\star$*-descriptively generalises* $\mathcal{L}$ if and only if, for every positive presentation of every language $L \in \mathcal{L}$, the sequence of hypotheses produced by the learner converges to a pattern δ that is descriptive of L with respect to the class $\mathrm{PAT}_\star$. A more formal definition of our model is given in Section 3.1.

The main difference between descriptive generalisation and related approaches (see, e. g., Arimura et al. (1994), Mukouchi (1994), Kobayashi and Yokomori (1995), Kobayashi and Yokomori (1997) and, indirectly, Jain and Kinber (2008)) is that we have a distinct split between a class $\mathcal{L}$ of languages to be inferred and an arbitrary class $\mathrm{PAT}_\star$ of pattern languages determining the set of admissible hypotheses. This leads to a compact and powerful model that yields interesting insights into the question of to which extent the generalisability of $\mathcal{L}$ depends on properties of $\mathcal{L}$ or of $\mathrm{PAT}_\star$. We discuss this topic in Section 3.2, and we demonstrate in Section 3.3 that descriptive generalisation can be interpreted as a natural instance of a very general and simple inference model which, to the best of our knowledge, has not been considered so far.

In Section 4, we investigate our model for a fixed and rich class $\mathrm{PAT}_\star$, namely the class of *terminal-free E-pattern languages*, i. e., the class of all pattern languages generated by patterns not containing any terminal symbols, where the empty word may be substituted for the variables in the pattern. Our studies reveal that, for this choice of $\mathrm{PAT}_\star$, descriptive generalisation and inductive inference from positive data are incomparable, and they show that there are major and natural classes of formal languages that can be descriptively generalised according to our model, but not precisely inferred in Gold's model. Technically, our decision to focus on terminal-free E-pattern languages leads to a number of substantial combinatorial challenges for pattern languages, and we present various respective insights and tools of intrinsic interest.

Due to space constraints, Sections 2.2 and 4 do not include any proofs of formal statements.

2 Preliminaries

This paper is largely self-contained. For language theoretic and recursion theoretic notations not explicitly defined, Rozenberg and Salomaa (1997) and Rogers (1992) can be consulted, respectively.

2.1 Definitions

Let $\mathbb{N} := \{0, 1, 2, 3, \dots\}$ and let ∞ denote infinity. The symbols $\subseteq$, $\subset$, $\supseteq$ and $\supset$ refer to subset, proper subset, superset and proper superset relation, respectively. The symbols $\mathcal{P}$ and $\setminus$ denote the power set and the set difference, respectively. For an arbitrary alphabet A, a *string* (over A) is a finite sequence of symbols from A, and λ stands for the *empty string*. The symbol A^+ denotes the set of all nonempty strings over A, and $A^* := A^+ \cup \{\lambda\}$.

For any alphabet A, a *language* L (over A) is a set of strings over A, i. e. $L \subseteq A^*$. A language L is *empty* if $L = \emptyset$; otherwise, it is *nonempty*. A *class* $\mathcal{L}$ of languages (over A) is a set of languages over A, i. e. $\mathcal{L} \subseteq \mathcal{P}(A^*)$. Let FIN_A denote the class of all finite languages over A.

For the *concatenation* of two strings w_1, w_2 we write $w_1 \cdot w_2$ or simply $w_1 w_2$. We say that a string $v \in A^*$ is a *factor* of a string $w \in A^*$ if there are $u_1, u_2 \in A^*$ such that $w = u_1 v u_2$. The notation $|K|$ stands for the size of a set K or the length of a string K; the term $|w|_a$ refers to the number of occurrences of the symbol a in the string w. For any $w \in \Sigma^*$ and any $n \in \mathbb{N}$, w^n denotes the *n-fold concatenation of* w, with $w^0 := \lambda$. Furthermore, we use $\cdot$ and the regular operations $*$ and $+$ on sets and strings in the usual way.

For any alphabets A, B, a *morphism* is a function $h : A^* \to B^*$ that satisfies $h(vw) = h(v)h(w)$ for all $v, w \in A^*$. Given morphisms $g : A^* \to B^*$ and $g : B^* \to C^*$ (for alphabets A, B, C), their *composition* $h \circ g$ is defined by $h \circ g(w) := h(g(w))$ for all $w \in A^*$.

A morphism $h : A^* \to B^*$ is said to be *nonerasing* if $h(a) \neq \lambda$ for all $a \in A$. For any string $w \in C^*$, where $C \subseteq A$ and $|w|_a \geq 1$ for every $a \in C$, the morphism $h : A^* \to B^*$ is called a *renaming (of w)* if $h : C^* \to B^*$ is injective and $|h(a)| = 1$ for every $a \in C$.

Let Σ be a (finite or infinite) alphabet of so-called *terminal symbols* (or: *letters*) and X an infinite set of *variables* with $\Sigma \cap X = \emptyset$. We normally assume $\{\mathtt{a}, \mathtt{b}, \ldots\} \subseteq \Sigma$ and $\{x_1, x_2, x_3 \ldots\} \subseteq X$. A *pattern* is a string over $\Sigma \cup X$, a *terminal-free pattern* is a string over X and a *word* is a string over Σ. The set of all patterns over $\Sigma \cup X$ is denoted by Pat_Σ. For any pattern α, we refer to the set of variables in α as $\mathrm{var}(\alpha)$, and to the set of terminal symbols in α as $\mathrm{symb}(\alpha)$.

A morphism $\sigma : (\Sigma \cup X)^* \to (\Sigma \cup X)^*$ is called *terminal-preserving* if $\sigma(a) = a$ for every $a \in \Sigma$. A terminal-preserving morphism $\sigma : (\Sigma \cup X)^* \to \Sigma^*$ is called a *substitution*.

The *NE-pattern language* $L_{\mathrm{NE},\Sigma}(\alpha)$ of a pattern $\alpha \in \mathrm{Pat}_\Sigma$ is given by

$$L_{\mathrm{NE},\Sigma}(\alpha) := \{\sigma(\alpha) \mid \sigma : (\Sigma \cup X)^* \to \Sigma^* \text{ is a nonerasing substitution}\},$$

and the *E-pattern language* $L_{\mathrm{E},\Sigma}(\alpha)$ of α is given by

$$L_{\mathrm{E},\Sigma}(\alpha) := \{\sigma(\alpha) \mid \sigma : (\Sigma \cup X)^* \to \Sigma^* \text{ is a substitution}\}.$$

Let ePAT_Σ denote the class of all E-pattern languages over Σ, and $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$ the class of all terminal-free E-pattern languages over Σ. Let $\mathrm{PAT}_{\star,\Sigma}$ be a class of NE-pattern languages or a class of E-pattern languages over Σ, and let $\mathrm{Pat}_{\star,\Sigma}$ be the corresponding class of generating patterns. If the correspondence is clear, we write $L(\alpha)$ instead of $L_{\mathrm{E},\Sigma}(\alpha)$ or $L_{\mathrm{NE},\Sigma}(\alpha)$ for any $\alpha \in \mathrm{Pat}_{\star,\Sigma}$.

Let $\mathrm{PAT}_{\star,\Sigma}$ be a class of NE-pattern languages or a class of E-pattern languages over Σ. We say that a pattern $\delta \in (\Sigma \cup X)^+$ is $\mathrm{PAT}_{\star,\Sigma}$*-descriptive* of a language $L \subseteq \Sigma^*$ if and only if $L(\delta) \in \mathrm{PAT}_{\star,\Sigma}$, $L(\delta) \supseteq L$, and there is no pattern α with $L(\alpha) \in \mathrm{PAT}_{\star,\Sigma}$ satisfying $L \subseteq L(\alpha) \subset L(\delta)$. Furthermore, $D_{\mathrm{PAT}_{\star,\Sigma}}(L)$ denotes the set of all patterns that are $\mathrm{PAT}_{\star,\Sigma}$-descriptive of L.

Let $\mathcal{L}$ be a class of languages over some alphabet A. Then $\mathcal{L}$ is said to be indexable provided that there exists an indexing $(L_i)_{i\in\mathbb{N}}$ of languages L_i such that, first, $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$ and, second, there exists a total computable function χ which uniformly decides the membership problem for $(L_i)_{i\in\mathbb{N}}$ – i. e., for every $w \in A^*$ and for every $i \in \mathbb{N}$, $\chi(w, i) = 1$ if and only if $w \in L_i$. In this case, we call $\mathcal{L} = (L_i)_{i\in\mathbb{N}}$ an *indexed family (of recursive languages)*. Of course, in this notation for an indexed family (which conforms with the use in the literature) the equality symbol "=" does not refer to an equality in the usual sense, but is merely a symbol indicating that $\mathcal{L}$ contains all languages in $(L_i)_{i\in\mathbb{N}}$ and vice versa.

2.2 Preliminary Results

Obviously, the definition of a descriptive pattern is based on the inclusion of pattern languages, which is an undecidable problem for both the full class of NE-pattern languages and the full class of E-pattern languages (cf. Jiang et al. (1995), Freydenberger and Reidenbach (2010)). A significant part of our subsequent technical considerations, however, is restricted to terminal-free E-pattern languages, where the inclusion problem is known to be decidable. This directly results from the following characterisation:

Theorem 1 (Jiang et al. (1994)) *Let $|\Sigma| \geq 2$. For every $\alpha, \beta \in X^+$, $L_{\mathrm{E},\Sigma}(\alpha) \subseteq L_{\mathrm{E},\Sigma}(\beta)$ holds if and only if there is a morphism $\phi : X^* \to X^*$ with $\phi(\beta) = \alpha$.*

Unfortunately, this problem is NP-complete:

Theorem 2 (Ehrenfeucht and Rozenberg (1979)) *Let Σ be an alphabet with $|\Sigma| \geq 2$. Then the inclusion problem for $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$ is NP-complete.*

On the other hand, in conjunction with Reidenbach and Schneider (2009), a recent result by Holub (2009) demonstrates that the equivalence problem can be decided in polynomial time:

Theorem 3 *There is a polynomial-time algorithm deciding, for any pair of terminal-free patterns α, β and for any alphabet Σ with $|\Sigma| \geq 2$, on whether $L_{\mathrm{E},\Sigma}(\alpha) = L_{\mathrm{E},\Sigma}(\beta)$.*

As shown by Freydenberger and Reidenbach (2009), not every language has an $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$- or an ePAT_Σ-descriptive pattern:

Theorem 4 *There is an infinite sequence $(\beta_n)_{n\geq 0}$ over X^+ such that, for every alphabet Σ with $|\Sigma| \geq 2$, $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L_\Sigma) = D_{\mathrm{ePAT}_\Sigma}(L_\Sigma) = \emptyset$ holds for the language $L_\Sigma := \bigcup_{n\geq 0} L_{\mathrm{E},\Sigma}(\beta_n)$.*

Note that L_Σ is an infinite language (and, in fact, an infinite union of languages from $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$). In contrast to this, Jiang et al. (1994) shows that every finite language has an ePAT_Σ-descriptive pattern. This is also true when considering $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns:

Proposition 5 *For every Σ with $|\Sigma| \geq 2$ and every finite nonempty $S \in \mathrm{FIN}_\Sigma$, $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(S) \neq \emptyset$, and a $\delta \in D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(S)$ can be effectively computed.*

While Proposition 5 proves the existence of an $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive pattern for every finite nonempty set, its proof makes use of a procedure that computes a descriptive pattern in a costly manner, since it solves the NP-complete inclusion problem (cf. Theorem 2) for an exponential number of patterns. Indeed, there is probably no algorithm that solves this problem in polynomial time:

Theorem 6 *Let Σ be an alphabet with $|\Sigma| \geq 2$. If $\mathrm{P} \neq \mathrm{NP}$, then there is no polynomial-time algorithm computing, for any finite set S of words, a pattern that is ePAT_Σ-descriptive of S.*

Theorem 6 addresses a problem left open by Jiang et al. (1994), and it provides a result that is stronger than the corresponding statement by Angluin (1980a) on NE-descriptive patterns. Since Theorem 6 can be proved using terminal-free patterns only, we can strengthen the corresponding result as follows:

Corollary 7 *Let Σ be an alphabet with $|\Sigma| \geq 2$. If $\mathrm{P} \neq \mathrm{NP}$, then there is no polynomial-time algorithm computing, for any finite set S of words, a pattern that is $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive of S.*

3 Inferring Descriptive Generalisations

In the present section, we formally introduce our notion of inferring descriptive generalisations, establish some of its basic properties (mainly by characterising, for any class of pattern languages determining the set of valid hypotheses, those indexed families that can be generalised in our model) and, finally, present a much more general inference paradigm that captures the essence of our approach. If we wish to compare Gold's well-known model of language identification in the limit from positive data (cf. Gold (1967)) with our model, then we refer to the former occasionally as LIM-TEXT. We use the same notation for the *class* of all classes of languages that can be *inferred* in that model; the meaning of this term shall therefore follow from the context.

3.1 The Inference Paradigm

We formalise our explanations on the model given in Section 1 as follows: For any alphabet Σ and any nonempty language $L \subseteq \Sigma^*$, we call a total function $t : \mathbb{N} \to \Sigma^*$ a *text* of L if and only if it satisfies $\{t(i) \mid i \in \mathbb{N}\} = L$. Moreover, for every text t and every $n \in \mathbb{N}$, t^n encodes the first n values of t in a single string, i. e. $t^n := t(1) \,\nabla\, t(2) \,\nabla\, t(3) \,\nabla\, \ldots \,\nabla\, t(n)$ with $\nabla \notin \Sigma$; additionally, we define $t[n] := \{t(i) \mid i \leq n\}$. Finally, $\mathrm{text}(L)$ denotes the set of all (computable and non-computable, repetitive and non-repetitive) texts of a language L.

Let $\mathcal{L}$ be a class of nonempty languages over an alphabet Σ, and let $\mathrm{PAT}_{\star,\Sigma}$ be a class of NE-pattern languages or a class of E-pattern languages over Σ. Then $\mathcal{L}$ is $\mathrm{PAT}_{\star,\Sigma}$*-descriptively generalisable* (or, if $\mathrm{PAT}_{\star,\Sigma}$ is understood, *(descriptively) generalisable* for short) if and only if there exists a computable function $S : (\Sigma \cup \{\nabla\})^* \to (\Sigma \cup X)^+$ such that, for every $L \in \mathcal{L}$ and for every $t \in \mathrm{text}(L)$, $S(t^n)$ is defined for every $n \in \mathbb{N}$, and there is a $\delta \in (\Sigma \cup X)^+$ with $\delta \in D_{\mathrm{PAT}_{\star,\Sigma}}(L)$ and there is an $m \in \mathbb{N}$ with $S(t^n) = \delta$ for every $n \geq m$. We call S a *(generalisation) strategy* and, for every $n \in \mathbb{N}$, $S(t^n)$ a *hypothesis* of S. The notation $\mathrm{DG}_{\mathrm{PAT}_{\star,\Sigma}}$ refers to the class of all classes of languages that are $\mathrm{PAT}_{\star,\Sigma}$-descriptively generalisable.

Consequently, and as already mentioned in Section 1, we have an inference model where the class to be inferred and the *hypothesis space* (we shall use this term in a rather informal manner for both the class $\mathrm{PAT}_{\star,\Sigma}$ and any set $\mathrm{Pat}_\star$ of patterns satisfying $\mathrm{PAT}_{\star,\Sigma} = \{L_\Sigma(\alpha) \mid \alpha \in \mathrm{Pat}_\star\}$) are entirely different objects. We feel that this feature precisely reflects our motivation as outlined in Section 1, and it establishes the difference of our approach to a number of related models.

3.2 Fundamental Insights into the Model

We now discuss some basic properties of descriptive generalisation without considering a specific class of pattern languages determining the hypothesis space. At first glance, the definitions of descriptive generalisation and of the LIM-TEXT model are closely related, and our first observation states that they are indeed equivalent if they are applied to any class of pattern languages:

Proposition 8 *Let $\mathrm{PAT}_{\star,\Sigma}$ be a class of pattern languages. Then $\mathrm{PAT}_{\star,\Sigma} \in \mathrm{LIM\text{-}TEXT}$ if and only if $\mathrm{PAT}_{\star,\Sigma} \in \mathrm{DG}_{\mathrm{PAT}_{\star,\Sigma}}$.*

Proof: Directly from the definitions of LIM-TEXT and $\mathrm{DG}_{\mathrm{PAT}_{\star,\Sigma}}$. ■

While descriptive generalisation and inductive inference from positive data, thus, seem to be very similar, there are major differences between these two models. In fact, there are classes that can be descriptively generalised, although neither the class nor the hypothesis space can be exactly inferred from positive data:

Proposition 9 *There exists a class $\mathcal{L}$ of languages and a class $\mathrm{PAT}_{\star,\Sigma}$ of pattern languages satisfying $\mathcal{L} \notin$ LIM-TEXT, $\mathrm{PAT}_{\star,\Sigma} \notin$ LIM-TEXT, and $\mathcal{L} \in \mathrm{DG}_{\mathrm{PAT}_{\star,\Sigma}}$.*

Proof: The statement follows from our Corollaries 26 and 31 in Section 4 and the fact that $\mathrm{ePAT}_{\mathrm{tf},\Sigma} \notin$ LIM-TEXT for $|\Sigma| = 2$ (cf. Reidenbach (2006)). ■

Since the definition of descriptive generalisation allows any class of pattern languages to be chosen as a hypothesis space, we can even devise a maximally powerful (yet utterly useless) generalisation strategy:

Proposition 10 *Let Σ be an alphabet. There exists a class $\mathrm{PAT}_{\star,\Sigma}$ of pattern languages such that every class $\mathcal{L}$ of languages over Σ satisfies $\mathcal{L} \in \mathrm{DG}_{\mathrm{PAT}_{\star,\Sigma}}$.*

Proof: Let $\mathrm{PAT}_{\star,\Sigma} := \{L_{\mathrm{E},\Sigma}(x_1)\}$. Since x_1 is $\mathrm{PAT}_{\star,\Sigma}$-descriptive of every language $L \subseteq \Sigma^*$, a strategy S that constantly outputs x_1 generalises $\mathcal{L}$. ■

Obviously, the substantial gap between the LIM-TEXT model and descriptive generalisation illustrated by Proposition 10 is based on a proof that uses a trivial notion of descriptiveness. In Section 4, we shall demonstrate that there are similarly deep differences between both models if a natural and nontrivial class of pattern languages, namely $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$, is used as admissible hypotheses for the generalisation process.

The main result of the present section is the following characterisation of descriptively generalisable indexed families of recursive languages. While our model as well as our studies in Section 4 consider descriptive generalisations of arbitrary classes of languages, this restriction facilitates an interesting comparison of our result to Angluin's characterisation of those indexed families that are inferrable in the LIM-TEXT model (see Angluin (1980b)). It is also worth noting that the subsequent argument cannot be based on strong insights into the descriptiveness of patterns, since we deal with arbitrary classes of pattern languages.

Theorem 11 *Let Σ be an alphabet, let $\mathcal{L} = (L_i)_{i\in\mathbb{N}}$ be an indexed family of nonempty recursive languages over Σ, and let $\mathrm{PAT}_{\star,\Sigma}$ be a class of pattern languages. $\mathcal{L} = (L_i)_{i\in\mathbb{N}} \in \mathrm{DG}_{\mathrm{PAT}_{\star,\Sigma}}$ if and only if there are effective procedures d and f satisfying the following conditions:*

(i) For every $i \in \mathbb{N}$, there exists a $\delta_{d(i)} \in D_{\mathrm{PAT}_{\star,\Sigma}}(L_i)$ such that d enumerates a sequence of patterns $d_{i,0}, d_{i,1}, d_{i,2}, \ldots$ satisfying, for all but finitely many $j \in \mathbb{N}$, $d_{i,j} = \delta_{d(i)}$.

(ii) For every $i \in \mathbb{N}$, f enumerates a finite set $F_i \subseteq L_i$ such that, for every $j \in \mathbb{N}$ with $F_i \subseteq L_j$, if $\delta_{d(i)} \notin D_{\mathrm{PAT}_{\star,\Sigma}}(L_j)$, then there is a $w \in L_j$ with $w \notin L_i$.

Proof: We begin with the *if* direction. In our proof, $F_i^{(m)}$ refers to the subset of F_i that is enumerated by f in $m \in \mathbb{N}$ steps of the computation.

We define a generalisation strategy S as follows: For any text t and for any $m \in \mathbb{N}$, when given t^m as an input, S outputs the pattern $d_{i,m}$, where $i \in \mathbb{N}$ is the smallest index satisfying: **(a)** $t[m] \subseteq L_i$ and **(b)** $F_i^{(m)} \subseteq t[m]$. If no such i exists, then S outputs $d_{0,0}$.

Since $\mathcal{L} = (L_i)_{i\in\mathbb{N}}$ is an indexed family, which means that the membership problem is uniformly decidable for all i and for all $w \in \Sigma^*$, and d and f are effective, it is obvious that S is computable and defined for every input t^m.

We now demonstrate that S $\mathrm{PAT}_{\star,\Sigma}$-descriptively generalises $\mathcal{L} = (L_i)_{i\in\mathbb{N}}$ if (i) and (ii) are satisfied. Thus, we choose an arbitrary $n \in \mathbb{N}$ and an arbitrary text t of L_n, and we show that S, when reading t, converges to a pattern that is $\mathrm{PAT}_{\star,\Sigma}$-descriptive of L_n. Before we start our actual reasoning, we determine a value $m_0 \in \mathbb{N}$ such that a number of vital parameters for the computation of $S(t^{m_0})$ have already stabilised: Let $m_1 \in \mathbb{N}$ be sufficiently large such that, for every $k \in \mathbb{N}$ with $k \leq n$ and $L_k \not\supseteq L_n$, $t[m_1]$ contains a word w satisfying $w \notin L_k$. The value m_1 must exists since $L_k \not\supseteq L_n$ and t is a text of L_n. Let $m_2 \in \mathbb{N}$ be sufficiently large such that, for every $k \in \mathbb{N}$ with $k \leq n$, $d_{k,m} = \delta_{d(k)}$ for every $m \geq m_2$. The value m_2 must exists due to (i). Let $m_3 \in \mathbb{N}$ be sufficiently large such that, for every $k \in \mathbb{N}$ with $k \leq n$, $F_k^{(m_3)} = F_k$. The value m_3 must exist since, according to (ii), F_k is finite. Let $m_4 \in \mathbb{N}$ be sufficiently large such that $F_n \subseteq t[m_4]$. The value m_4 must exist since t is a text of L_n and, according to (ii), $F_n \subseteq L_n$. Then $m_0 := \max\{m_1, m_2, m_3, m_4, n\}$.

Referring to these definitions, our proof of the *if* direction is based on the following Claims:

Claim 1. For every $m \geq m_0$, n satisfies $t[m] \subseteq L_n$, and $F_n^{(m)} \subseteq t[m]$.

Proof (Claim 1). The first part of the statement holds since t is a text of L_n; the second part holds because of $m \geq m_0 \geq m_4$. ■ (Claim 1)

Claim 2. For every $m \geq m_0$, $S(t^m) \in D_{\mathrm{PAT}_{\star,\Sigma}}(L_n)$.

Proof (Claim 2). Let $S(t^m) = d_{k,m}$. By definition, S outputs the pattern $d_{k,m}$ for the smallest index $k \leq m$ satisfying conditions (a) and (b), or it outputs the auxiliary hypothesis $d_{0,0}$ if such a k does not exist. Due to Claim 1, and since $m \geq m_0$, we know that there exists at least one index (namely n) satisfying (a) and (b) for t^m. Thus, $m \geq m_0 \geq n$ implies that S does not choose to output its auxiliary hypothesis. Therefore, the following statements hold true for k: **(1)** $k \leq n$ (since S outputs $d_{k,m}$ for the smallest k satisfying conditions (a) and (b) of the definition of S); **(2)** $d_{k,m} = \delta_{d(k)}$ (because of $m \geq m_0 \geq m_2$ in conjunction with statement (1)); **(3)** $t[m] \subseteq L_k$ (because of condition (a)); **(4)** $F_k \subseteq t[m] \subseteq L_n$ (because of $m \geq m_0 \geq m_3$ in conjunction with statement (1), and due to condition (b)).

Now assume to the contrary that $S(t^m) = d_{k,m} \notin D_{\mathrm{PAT}_{\star,\Sigma}}(L_n)$. Due to statement (2), this means that $\delta_{d(k)} \notin D_{\mathrm{PAT}_{\star,\Sigma}}(L_n)$. Then statement (4) and condition (ii) of the Theorem imply that there exists a word $w \in L_n \setminus L_k$ which, due to $m \geq m_0 \geq m_1$ in conjunction with statement (1), satisfies $w \in t[m]$. Hence, $t[m] \not\subseteq L_k$. This contradicts statement (3). ■ (Claim 2)

Claim 3. There is a pattern δ and an $m' \geq m_0$ such that, for every $m \geq m'$, $S(t^m) = \delta$.

Proof (Claim 3). Due to statement (1) in the proof of Claim 2 and $m \geq m_0 \geq m_2$, there is only a finite number of possible hypotheses – namely $\delta_{d(0)}, \delta_{d(1)}, \ldots, \delta_{d(n)}$ – that S can output when reading t^m. Therefore, it is sufficient to show that a hypothesis, once it has been discarded, is not chosen by S anymore. More precisely, we prove that if, for an $l_0 \geq m$, $S(t^{l_0}) = \delta_{d(k)}$ and $S(t^{l_0+1}) \neq \delta_{d(k)}$, then, for every $l \geq l_0 + 1$, there exists a $k' \neq k$ with $S(t^l) = \delta_{d(k')}$.

Since $l_0 \geq m \geq m_3$, $S(t^{l_0}) = \delta_{d(k)}$ implies **(A)** $t[l_0] \subseteq L_k$ and **(B)** $F_k \subseteq t[l_0]$. By definition, $t[l_0 + 1] \supseteq t[l_0]$, and therefore (B) is satisfied for t^{l_0+1}, too. Thus, the only event that can trigger a change of the hypothesis when extending t^{l_0} to t^{l_0+1} is $t[l_0+1] \not\subseteq L_k$; this implies $M \not\subseteq L_k$ for all supersets M of $t[l_0 + 1]$. Hence, for every $l \geq l_{0+1}$, there is a $k' \neq k$ with $S(t^l) = \delta_{d(k')}$. ■ (Claim 3)

To summarise, Claim 3 shows that S converges when reading t to a pattern δ, and Claim 2 demonstrates that δ is $\mathrm{PAT}_{\star,\Sigma}$-descriptive of L_n. This concludes the proof of the *if* direction.

We continue with the *only if* direction. Hence, let S be a computable generalisation strategy that $\mathrm{PAT}_{\star,\Sigma}$-descriptively generalises $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$, i. e., for every i and for every text t of L_i, S converges to a pattern that is $\mathrm{PAT}_{\star,\Sigma}$-descriptive of L_i. We show that this implies the existence of effective procedures d and f satisfying conditions (i) and (ii).

Since $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ is an indexed family, there is an effective procedure enumerating, for every $i \in \mathbb{N}$, all words $w_{i,0}, w_{i,1}, w_{i,2}, \ldots$ in L_i. Furthermore, we can use this to define a second effective procedure which enumerates, for every $i \in \mathbb{N}$, all finite sequences $s_{i,0}, s_{i,1}, s_{i,2}, \ldots$ of words in L_i. Note that each sequence $s_{i,j}$, $j \in \mathbb{N}$, may contain repetitions of words. Furthermore, if L_i is finite, we can nevertheless easily make sure that the output of the above procedures is infinite for every i.

We now give a procedure that defines the behaviour of d and f:

Procedure SIM_S

Let $i \in \mathbb{N}$, and let $w_{i,0}, w_{i,1}, w_{i,2}, \ldots$ and $s_{i,0}, s_{i,1}, s_{i,2}, \ldots$ be as given above. Go to Stage 0.

Stage 0. Define $t_0 := w_{i,0}$, $F_i := \{w_{i,0}\}$. Define $x := 0$ and $d_{i,x} := S(t_0)$. Go to Stage 1.

Stage n ($n \geq 1$). For every $j = 0, 1, 2, \ldots$ proceed as follows: Consider $s_{i,j} = (\widehat{w}_{j,0}, \widehat{w}_{j,1}, \ldots, \widehat{w}_{j,y})$, $y \in \mathbb{N}$, and define $t'_j := \widehat{w}_{j,0} \nabla \widehat{w}_{j,1} \nabla \ldots \nabla \widehat{w}_{j,y}$. Define $x := x + 1$ and $d_{i,x} = S(t_{n-1} \nabla t'_j)$. If $S(t_{n-1} \nabla t'_j) \neq S(t_{n-1})$, then define $t_n := t_{n-1} \nabla t'_j \nabla w_{i,n}$, $F_i := F_i \cup \{\widehat{w}_{j,0}, \widehat{w}_{j,1}, \ldots, \widehat{w}_{j,y}, w_{i,n}\}$, and go to Stage $n + 1$.

Since S and the procedures enumerating the $w_{i,j}$ and $s_{i,j}$, $i, j \in \mathbb{N}$, are computable, the same holds for SIM_S. Consequently, effective procedures d and f which, for all $i \in \mathbb{N}$, uniformly produce sequences $d_{i,0}, d_{i,1}, d_{i,2}, \ldots$ and enumerate F_i, respectively, can be directly derived from SIM_S.

We now show that d and f satisfy conditions (i) and (ii). Our corresponding reasoning makes use of the following fact:

Claim 4. For every $i \in \mathbb{N}$ there exists an n_0 such that procedure SIM_S, when given input i, enters Stage n_0, but it does not enter Stage $n_0 + 1$.

Proof (Claim 4). Assume to the contrary that procedure SIM_S enters an infinite number of stages. This implies that S does not converge to a fixed pattern, since SIM_S goes to the next stage if and only if S changes its hypothesis for the given input. However, since all considered words are contained in L_i, each transition from Stage n to Stage $n+1$ adds the word $w_{i,n}$ to t_n, and $\{w_{i,j} \mid j \in \mathbb{N}\} = L_i$, the string $\lim_{n \to \infty} t_n$ is an encoding of a text t of L_i. Since S $\mathrm{PAT}_{\star,\Sigma}$-descriptively generalises $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$, this means that S, when reading t, must converge to a pattern. This is a contradiction. ■ (Claim 4)

By definition, for every $i \in \mathbb{N}$, SIM_S produces an infinite sequence of patterns. It outputs a pattern $d_{i,x+1}$ that differs from $d_{i,x}$ only if it moves from one stage to another. Thus, due to Claim 4 and for the corresponding $x_0 \in \mathbb{N}$, the sequence of patterns $d_{i,x_0}, d_{i,x_0+1}, d_{i,x_0+2}, \ldots$ produced in Stage n_0 satisfies, for every $j \in \mathbb{N}$, $d_{i,x_0+j} = d_{i,x_0}$. Furthermore, due to fact that the constructed input is a text of L_i, S needs to converge to a $\mathrm{PAT}_{\star,\Sigma}$-descriptive pattern of L_i. This implies that $d_{i,x_0} = \delta_{d(i)}$ for a $\delta_{d(i)} \in D_{\mathrm{PAT}_{\star,\Sigma}}(L_i)$. Consequently, the sequence of patterns $d_{i,0}, d_{i,1}, d_{i,2}, \ldots$ satisfies condition (i).

SIM_S adds a finite number of words to F_i if and only if it moves to the next stage. Hence, Claim 4 shows that each F_i is finite. Now assume to the contrary that, for an $i \in \mathbb{N}$, F_i does not satisfy condition (ii), i. e., there exists a $j \in \mathbb{N}$ with $F_i \subseteq L_j$, $\delta_{d(i)} \notin D_{\mathrm{PAT}_{\star,\Sigma}}(L_j)$ and $L_j \subseteq L_i$. Let $t_{<j>}$ be an arbitrary text of L_j. Since $F_i \subseteq L_j$ and t_{n_0-1} encodes the words in F_i, for every $m \in \mathbb{N}$, $t_{n_0-1} \triangledown t^m_{<j>}$ is an encoding of initial values of a text of L_j. Thus, for $m \to \infty$, S must converge when reading $t_{n_0-1} \triangledown t^m_{<j>}$ to a pattern that is $\mathrm{PAT}_{\star,\Sigma}$-descriptive of L_j. According to Claim 4, when t_{n_0-1} is continued with the encoding of any finite sequence of words from L_i, SIM_S does not leave Stage n_0. Since $L_j \subseteq L_i$, this implies that SIM_S does not leave Stage n_0 for t_{n_0-1} being continued with the encoding of any finite sequence of initial values of $t_{<j>}$. Therefore S converges, when given $t_{n_0-1} \triangledown t^m_{<j>}$ for $m = 0, 1, 2, \ldots$, by definition to $\delta_{d(i)}$. This contradicts $\delta_{d(i)} \notin D_{\mathrm{PAT}_{\star,\Sigma}}(L_j)$. Consequently, F_i satisfies condition (ii), and this concludes the proof of the *only if* direction.

Hence, $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ is $\mathrm{PAT}_{\star,\Sigma}$-descriptively generalisable if and only if there are effective procedures d and f satisfying conditions (i) and (ii). ■

As briefly mentioned above, Theorem 11 shows natural connections to the seminal characterisation of learnable indexed families given by Angluin (1980b), and therefore it is not surprising that some elements of our proof do not need to differ from hers. Most of these similarities result from the fact that each successful inductive inference process requires the existence of so-called *locking sequences* (see Lange et al. (2008) for a detailed discussion), and this is reflected by Angluin's *telltale* T_i and our comparable concept F_i. Nevertheless, there are crucial differences between the two characterisations. First, we need to define an enumeration of an appropriate subset of our hypothesis space (this is done by the procedure d), whereas this is automatically given in Angluin's model. In this context, it is important to note that we have to attune the set F_i to the pattern $\delta_{d(i)}$, $i \in \mathbb{N}$, which leads to d and f being defined by the same procedure SIM_S. Second, while Angluin's T_i must, for *every* j with $L_j \subset L_i$, contain a word from $L_i \setminus L_j$, our equivalent F_i only needs to do so if $\delta_{d(i)}$ is not an acceptable hypothesis for L_j. This fits with the requirement of inductive inference from positive examples to distinguish between *all* languages L_i and L_j with $L_i \neq L_j$, whereas descriptive generalisation only has to distinguish between some of them, and this requisite might be *asymmetric*, i. e., a strategy S might have to discover that a text of a language L_i is not a text of a language L_j, but it might not need to figure out that a text of L_j is not a text of L_i. The explanation of why descriptive generalisation, in general, is more powerful than inductive inference from positive data directly follows from this observation; further considerations on this topic are given in Section 3.3. Thirdly, and finally, the strategy S we deploy in our proof is, in a sense, not optimal, as it might discard a correct hypothesis – i. e. pattern $\delta_{d(j)}$ that incidentally is descriptive of the language L_i the text of which is read – simply because L_i contains a word that is not contained in L_j.

Our generic strategy S of course is not very efficient; furthermore, it has the bothersome property described above. However, it is worth mentioning that S does not test whether the given words are contained in the language of the hypothesis pattern, and it does not check the inclusion of pattern languages, either. Thus, it circumvents two decision problems that, for many natural classes of pattern languages, are known to be NP-complete or even undecidable (see, e. g., Angluin (1980a) and Freydenberger and Reidenbach (2010)), although these decision problems are essential elements of the definition of descriptiveness. Instead, S infers descriptive patterns purely based on membership tests for the languages in the indexed family. Thus, if indexed families with a fast membership test are to be generalised, then our strategy raises hope that it might be possible to do this efficiently in spite of using a hypothesis space with an NP-complete membership test. On the other hand, it might be difficult to find rich classes of pattern languages where the procedure d introduced by Theorem 11 is efficient (even though it should normally be possible to devise a d that, for every $i \in \mathbb{N}$, directly outputs the pattern $\delta_{d(i)}$ instead of enumerating the sequence $d_{i,j}$). This expectation is substantiated by Theorem 4.2 in Angluin (1980a) and our Theorem 6 and Corollary 7 given in Section 2.

3.3 A More General View

While an application of Theorem 11 might require profound knowledge on the descriptiveness of patterns, a closer look confirms our above remark that the actual characterisation and its proof do not at all. More precisely, neither the Theorem nor our reasoning deal with the properties of the descriptive patterns $\delta_{d(i)}$, $i \in \mathbb{N}$, but they merely make use of a notion of the *validity* of a hypothesis for a given language, i. e., a hypothesis is acceptable for a language if it is descriptive, but we do not check for descriptiveness. This view is quite convenient to study the difference between descriptive generalisation and inductive inference from positive data. In the LIM-TEXT model when applied to indexed families, a hypothesis i – i. e., the index of the language L_i – is valid for a language L_j, $j \neq i$, if and only if the hypothesis j is valid for the language L_i (if and only if $L_i = L_j$). In our model, this symmetry does not necessarily exist, as demonstrated by the following example:

Example 12 *Let* $\Sigma := \{\mathtt{a}, \mathtt{b}\}$*. Let* $L_1 := \{\mathtt{a\,b\,a\,b\,a}, \mathtt{b\,a\,b\,a\,b}\}$ *and* $L_2 := \{\mathtt{a\,b\,a\,b\,a}, \mathtt{b\,a\,b\,a\,b}, \mathtt{a\,b\,a\,a\,b\,a}\}$*. We state without proof that* $\delta_1 := x_1\,\mathtt{a\,b\,a\,b}\,x_2$ *is* ePAT_Σ*-descriptive of* L_1 *and* $\delta_2 := x_1 x_2 x_1 x_2 x_1$ *is* ePAT_Σ*-descriptive of* L_2*. While* δ_2 *is also* ePAT_Σ*-descriptive of* L_1*,* δ_1 *is not* ePAT_Σ*-descriptive of* L_2*. Hence, a strategy* S *that* ePAT_Σ*-descriptively generalises a class including* L_1 *and* L_2 *can output* δ_1 *or* δ_2 *when reading a text for* L_1*, but it must not output* δ_1 *when reading a text for* L_2*.*

Referring to this phenomenon and restricted to indexed families, we can now give a much more general model of inference than the one of descriptive generalisation, and we can still characterise those indexed families that can be inferred according to this model in exactly the same way as we have done in Theorem 11. Hence, let $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ be an indexed family. Furthermore, for any $i \in \mathbb{N}$, let HYP be a function that maps i to a subset of $\mathbb{N}$ that consists of all *valid hypotheses* for L_i. Here it is important to note that the numbers in $\mathrm{HYP}(i)$ do normally not refer to indices of the indexed family $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$; e. g., in our model of descriptive generalisation they would stand for indices in an arbitrary enumeration of a set of patterns. We then say that $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ is *inductively inferrable with hypotheses validity relation* HYP if and only if there exists a computable function $S : (\Sigma \cup \{\nabla\})^* \to \mathbb{N}$ such that, for every $i \in \mathbb{N}$ and for every $t \in \mathrm{text}(L_i)$, $S(t^n)$ is defined for every $n \in \mathbb{N}$ and there is a $j \in \mathrm{HYP}(i)$ and there is an $m \in \mathbb{N}$ with $S(t^n) = j$ for every $n \geq m$.

Our notion of descriptive generalisation demonstrates that there are natural instances of the model of inductive inference with hypotheses validity relation HYP. Nevertheless, to the best of our knowledge, its properties have not been explicitly studied so far.

As announced above, we now rephrase Theorem 11 so that it characterises those indexed families that are inductively inferrable with hypotheses validity relation HYP:

Theorem 13 *Let* Σ *be an alphabet, let* $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ *be an indexed family of nonempty languages over* Σ*, and let* $\mathrm{HYP} : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ *be a function.* $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ *is inductively inferrable with hypotheses validity relation* HYP *if and only if there are effective procedures* h *and* f *satisfying the following conditions:*

(i) For every $i \in \mathbb{N}$*, there exists a* $\eta_i \in \mathrm{HYP}(i)$ *such that* h *enumerates a sequence of natural numbers* $i_0, i_1, i_2, \ldots$ *satisfying, for all but finitely many* $k \in \mathbb{N}$*,* $i_k = \eta_i$*.*

(ii) For every $i \in \mathbb{N}$*,* f *enumerates a finite set* $F_i \subseteq L_i$ *such that, for every* $j \in \mathbb{N}$ *with* $F_i \subseteq L_j$*, if* $\eta_i \notin \mathrm{HYP}(j)$*, then there is a* $w \in L_j$ *with* $w \notin L_i$*.*

Proof: Minor and straightforward editing of the proof of Theorem 11 – mainly substituting h for d, i_k for $d_{i,k}$, η_i for $\delta_{d(i)}$, and $\mathrm{HYP}(i)$ for $D_{\mathrm{PAT}_{\star,\Sigma}}(L_i)$ – turns it into a reasoning suitable for Theorem 13. ■

To conclude this section on basic properties of our model, we wish to mention that descriptive generalisation can alternatively be interpreted as inductive inference of classes of pattern languages from *partial texts*. Hence, we can understand any language L_i as a tool to define texts that do not contain all words in $L(\delta_{d(i)})$, but nevertheless can be used to infer $\delta_{d(i)}$. Within the scope of the present paper, we do not explicitly discuss such a view, but we expect that it might be a worthwhile topic for further studies. We anticipate that its analysis might involve substantial conceptual challenges that cannot be solved using established insights into related approaches (see Fulk and Jain (1996)).

4 Inferring $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-Descriptive Patterns

We now study our model for a fixed hypothesis space, namely the class $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$. The decidability of the inclusion problem for this class (see Theorem 1) allows us to develop a set of powerful tools.

This section is divided into three parts. In the first part, we consider some questions on the existence of $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns for various classes of languages and develop a set of tools in order to simplify proofs on the existence and nonexistence of $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns.

The second part deals with a generalisation strategy that is based on the procedure that is described in Proposition 5, which we deem so natural that we call it the canonical strategy Canon for $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive generalisations. Most importantly, we give a characterisation of the class $\mathcal{TSL}_\Sigma$ of languages that can be descriptively generalised with Canon.

In the final part of this section, we examine the relationship of various classes of languages to $\mathcal{TSL}_\Sigma$ in order to gain further insights into $\mathrm{DG}_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}$ and the power of Canon.

4.1 Basic tools

Before we proceed to an examination of $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive generalisation in the next part of this section, we develop some tools and techniques that simplify the work with $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns, and gather some results on the existence and nonexistence of such patterns for some classes of languages. We begin with the following result:

Lemma 14 *Let Σ be an alphabet with $|\Sigma| \geq 2$, and let $L_1, L_2 \subseteq \Sigma^*$ with $L_1 \supseteq L_2$. If there is a $\delta \in D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L_2)$ with $L_{\mathrm{E},\Sigma}(\delta) \supseteq L_1$, then $\delta \in D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L_1)$.*

This observation might seem to be elementary, but together with Lemma 17, it forms the fundament of the proof of almost every result in this section. The technical base of that Lemma derives from a phenomenon that often arises when dealing with $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns. We consider the following example:

Example 15 *Let $\Sigma := \{\mathtt{a}, \mathtt{b}\}$ and let $L_1 := \{\mathtt{a}^2\}$, $L_2 := \{(\mathtt{a}\,\mathtt{b}^1\,\mathtt{a}\;\mathtt{a}\,\mathtt{b}^2\,\mathtt{a} \ldots \mathtt{a}\,\mathtt{b}^n\,\mathtt{a})^2 \mid n \geq 2\}$, and $L_3 := L_{\mathrm{E},\Sigma}(x_1^2) \setminus \{\mathtt{a}^2, \mathtt{b}^2\}$. It is easy to see that all three languages are included in $L_{\mathrm{E},\Sigma}(x_1^2)$. However, in addition to this, for every $\alpha \in X^+$ with $L_{\mathrm{E},\Sigma}(\alpha) \supseteq L_i$ (with $1 \leq i \leq 3$), $L_{\mathrm{E},\Sigma}(\alpha) \supseteq L_{\mathrm{E},\Sigma}(x_1^2)$ holds as well. For L_1, this is obvious. For L_2, assume that $L_{\mathrm{E},\Sigma}(\alpha) \supseteq L_2$ for some $\alpha \in X^+$, let $n := |\operatorname{var}(\alpha)|$ and $w = (\mathtt{a}\,\mathtt{b}^1\,\mathtt{a}\;\mathtt{a}\,\mathtt{b}^2\,\mathtt{a} \ldots \mathtt{a}\,\mathtt{b}^n\,\mathtt{a})^2 \in L_2$, and choose any morphism ϕ with $\phi(\alpha) = w$. As w contains n distinct factors of the form $\mathtt{a}\,\mathtt{b}^+\,\mathtt{a}$, each occurring exactly twice, there must be an $x \in \operatorname{var}(\alpha)$ that contains at least one complete occurrence of such a segment, which implies $|\alpha|_x \in \{1, 2\}$. In both cases, we can construct a morphism ψ with $\psi(\alpha) = x_1^2$ (by mapping x to x_1 or x_1^2 and erasing all other variables), which (according to Theorem 1) leads to $L_{\mathrm{E},\Sigma}(\alpha) \supseteq L_{\mathrm{E},\Sigma}(x_1^2)$. Finally, as $L_3 \supset L_2$, this also proves the claim for L_3.*

As $L_{\mathrm{E},\Sigma}(x_1^2)$ and all three L_i have exactly the same superpatterns, we are able to conclude that, for every $i \in \{1, \ldots, 3\}$, $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L_{\mathrm{E},\Sigma}(x_1^2)) = D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L_i)$. Although the four languages might seem rather different, they have exactly the same sets of $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns.

When generalising languages using $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns, every language has a certain superset that is covered by every descriptive generalisation of this language, and cannot be avoided. In order to formalise this line of reasoning (and in order to use this phenomenon), we introduce the set of *superpatterns* $\mathrm{Super}(L)$, and the *superpattern hulls* $\text{S-Hull}_\Sigma(L)$, which are defined as

$$\mathrm{Super}(L) := \{\alpha \in X^+ \mid \text{for every } w \in L, \text{ there is a morphism } \phi \text{ with } \phi(\alpha) = w\},$$
$$\text{S-Hull}_\Sigma(L) := \bigcap_{\alpha \in \mathrm{Super}(L)} L_{\mathrm{E},\Sigma}(\alpha)$$

for all alphabets Σ, Σ' and any language $L \subseteq (\Sigma')^*$. Note that, by Theorem 1, for every pair of patterns $\alpha, \beta \in X^+$ and every Σ with $\Sigma \geq 2$, $L_{\mathrm{E},\Sigma}(\alpha) \subseteq L_{\mathrm{E},\Sigma}(\beta)$ if and only if $\beta \in \mathrm{Super}(L_{\mathrm{E},\Sigma}(\alpha))$ if and only if $\beta \in \mathrm{Super}(\{\alpha\})$. This allows us to state the following corollary:

Corollary 16 *Let Σ, Σ' be alphabets with $|\Sigma|, |\Sigma'| \geq 2$. Then $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L) = D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma'}}(L)$ for every $L \subseteq (\Sigma \cap \Sigma')^*$.*

Although $\mathrm{Super}(L)$ and $\text{S-Hull}_\Sigma(L)$ might appear to be rather simple concepts, they can be used to establish most of the results in this section. Using Lemma 14, we can develop one of our main tools:

Lemma 17 *Let Σ be an alphabet with $|\Sigma| \geq 2$. For every $L \subseteq \Sigma^*$, $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L) = D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(\text{S-Hull}_\Sigma(L))$.*

In a sense, $\text{S-Hull}_\Sigma(L)$ captures the whole essence of L with respect to $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive patterns, as every $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive generalisation of L is unable to distinguish between these two languages. This is illustrated by the following example:

Example 18 *Let $|\Sigma| \geq 2$ and define $L := L_{\mathrm{E},\Sigma}(x_1^2) \cup L_{\mathrm{E},\Sigma}(x_1^3)$. Furthermore, let*

$$\begin{array}{lllll} \delta_1 := x_1^2 x_2^3, & \delta_2 := x_1 x_2 x_1 x_2^2, & \delta_3 := x_1 x_2^2 x_1 x_2, & \delta_4 := x_1 x_2^3 x_1, & \delta_5 := x_1 x_2^2 x_1^2, \\ \delta_6 := x_1 x_2 x_1 x_2 x_1, & \delta_7 := x_1 x_2 x_1^2 x_2, & \delta_8 := x_1^2 x_2^2 x_1, & \delta_9 := x_1^2 x_2 x_1 x_2, & \delta_{10} := x_1^3 x_2^2. \end{array}$$

Recalling Theorem 1, it is easy to see that, for every $\alpha \in \mathrm{Super}(L)$, there is a δ_i, $1 \leq i \leq 10$, with $L_{\mathrm{E},\Sigma}(\alpha) \supseteq L_{\mathrm{E},\Sigma}(\delta_i)$ (as, for every α, there must be morphisms mapping α to both x_1^2 and x_1^3). By a *convention common in the literature, all patterns are given in* canonical form *(cf. Reidenbach and Schneider (2009)), where variables names are introduced in increasing lexicographic order.*

This example illustrates two important phenomena. First, note that $\delta_i \in D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(L)$ for $1 \leq i \leq 10$, and for every $\delta \in D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(L)$, there is a δ_i with $L_{\mathrm{E},\Sigma}(\delta) = L_{\mathrm{E},\Sigma}(\delta_i)$, but $L_{\mathrm{E},\Sigma}(\delta) \neq L_{\mathrm{E},\Sigma}(\delta_j)$ for every $j \neq i$. Thus, L has ten distinct $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$*-descriptive patterns.*

Second, the previous observation leads to $\text{S-Hull}_\Sigma(L) = \bigcap_{i=1}^{10} L_{\mathrm{E},\Sigma}(\delta_i)$*. For every $n \geq 2$, there are $j, k \geq 0$ with $n = 2j + 3k$, and therefore,* $\text{S-Hull}_\Sigma(L) \supseteq \bigcup_{n=2}^{\infty} L_{\mathrm{E},\Sigma}(x_1^n)$*. Thus, every* $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$*-descriptive generalisation of L is unable to exclude any language $L_{\mathrm{E},\Sigma}(x_1^n)$ with $n \geq 2$. In this sense,* $\text{S-Hull}_\Sigma(L)$ *provides information on the coarseness of all descriptive generalisations.*

Observe that L in the previous example is a finite union of languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$ that has a descriptive pattern, and recall that, according to Proposition 5, every finite set of words has an $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$-descriptive pattern, while (by Theorem 4), there are infinite unions of languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$ that have no descriptive pattern.

Using Lemma 17, we can extend Proposition 5 to show that not only every finite set of words, but every finite union of languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$ has an $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$-descriptive pattern:

Proposition 19 *Let Σ be an alphabet with $|\Sigma| \geq 2$, let $A = \{\alpha_1, \ldots, \alpha_n\} \subset X^+$ and let $L = \bigcup_{i=1}^n L_{\mathrm{E},\Sigma}(\alpha_i)$. Then $D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(L) \neq \emptyset$.*

Basically, Example 18 and Proposition 19 are based on the fact that words in languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$ and the generating patterns of these languages can often be used interchangeably by defining a morphism that maps the words back to their generating pattern. We proceed to develop this approach into another tool that allows us to make further statements on the (non-)existence of $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$-descriptive patterns. Let $\nu : \Sigma^* \to X^*$ an arbitrary renaming. We define $\text{V-Hull}_\Sigma(L) := \bigcup_{w \in L} L_{\mathrm{E},\Sigma}(\nu(w))$. Like $\text{S-Hull}_\Sigma(L)$, $\text{V-Hull}_\Sigma(L)$ is equivalent to L with respect to Super and $D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}$:

Lemma 20 *Let Σ be an alphabet, $|\Sigma| \geq 2$. For every L over Σ,* $\mathrm{Super}(L) = \mathrm{Super}(\text{V-Hull}_\Sigma(L))$*, and* $D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(L) = D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(\text{V-Hull}_\Sigma(L))$.

This leads us to the following insight into the existence of $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$-descriptive patterns for infinite unions of languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$:

Proposition 21 *Let $|\Sigma| \geq 2$. Then there is a set of patterns $A \subset \{x_1, x_2\}^+$ such that no pattern is* $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$*-descriptive of $\bigcup_{\alpha \in A} L_{\mathrm{E},\Sigma}(\alpha)$.*

Thus, unlike in the case of *finite* unions of languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$ (cf. Proposition 19), even restricting the number of variables in the generating patterns does not ensure that *infinite* unions of languages from $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$ have a descriptive pattern. The renaming ν that maps terminals to variables can also be used to obtain the following technical result:

Lemma 22 *Let Σ be an alphabet with $|\Sigma| \geq 2$. For every nonempty language $L \subseteq \Sigma^*$,* $\text{S-Hull}_\Sigma(L)$ *is infinite.*

This insight shall be used in Section 4.3. We conclude the present part of Section 4 with a short remark illustrating that there are finite classes of languages which are not contained in $\mathrm{DG}_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}$:

Proposition 23 *Let Σ be an alphabet, $|\Sigma| \geq 2$. There exists a class $\mathcal{L}$ of nonempty languages over Σ with $|\mathcal{L}| = 1$ and $\mathcal{L} \notin \mathrm{DG}_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}$.*

4.2 The Canonical Strategy and Telling Sets

According to Proposition 5, every finite set has a computable $\mathrm{ePAT}_{\mathtt{tf},\Sigma}$-descriptive pattern. We consider it the canonical strategy of descriptive inference on any text t of a given language L to compute a descriptive pattern of every initial segment t^n, in the hope that the hypothesises will converge to a pattern that is descriptive of L. As evidenced by the language $L := L_{\mathrm{E},\Sigma}(x_1^2) \cup L_{\mathrm{E},\Sigma}(x_1^3)$ (cf. Example 18), there are languages with more than one descriptive pattern. Furthermore, this applies also to finite languages, as for the set $S := \{\mathtt{a}^2, \mathtt{b}^3\}$ (for arbitrary letters $\mathtt{a}, \mathtt{b} \in \Sigma$), $D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(S) = D_{\mathrm{ePAT}_{\mathtt{tf},\Sigma}}(L)$ holds. Although S already contains all the information that is needed to compute a descriptive generalisation of L, the six distinct patterns δ_1 to δ_6 from Example 18 are all valid hypothesises. In order to allow our strategy to converge to one single hypothesis, we impose a total and well-founded order $<_{\mathrm{LLO}}$ on X^+ and let our strategy return the $<_{\mathrm{LLO}}$-minimal hypothesis.

Let $<_{\mathrm{LLO}}$ denote the length-lexicographic order[1] on X^+. Note that $<_{\mathrm{LLO}}$ is total and does not contain infinite decreasing chains. Thus, every set has exactly one element that is minimal with respect to $<_{\mathrm{LLO}}$.

[1] I. e., $\alpha <_{\mathrm{LLO}} \beta$ if $|\alpha| < |\beta|$, or if $|\alpha| = |\beta|$, and α precedes β in the lexicographic order.

The strategy $\mathrm{Canon} : (\Sigma \cup \{\nabla\})^* \to (\Sigma \cup X)^+$ is defined by, for every text t,

$$\mathrm{Canon}(t^n) := \delta, \text{ where } \delta \in D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(t[n]) \text{ and } \delta <_{\mathrm{LLO}} \gamma \text{ for every other } \gamma \in D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(t[n]).$$

The computability of Canon follows immediately from the proof of Proposition 5, as all that remains is to sort the finite search space by $<_{\mathrm{LLO}}$. We say that Canon *converges* on a text $t \in \mathrm{text}(L)$ (of some language L over some alphabet Σ) if there is a pattern $\alpha \in X^+$ with $\mathrm{Canon}(t^n) = \alpha$ for all but finitely many values of n. If, in addition to this, $\alpha \in D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L)$, Canon is said to *converge correctly* on t. Now, when considering the languages L and S given in the example above, for every text $t \in \mathrm{text}(L)$, there is an $n \geq 0$ with $S \subseteq t[n]$. From this point on, $\mathrm{Canon}(t[n])$ will return the pattern $\delta_{10} = x_1^3 x_2^2$, as δ_{10} is an element of $(D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(S) \cap D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L))$ and the $<_{\mathrm{LLO}}$-minimum of the canonical forms of the δ_i. This phenomenon leads to the definition of what we call *telling sets*, which are of crucial importance for the study of descriptive generalisability with the strategy Canon:

Definition 24 *Let $L \subseteq \Sigma^*$. A finite set $S \subseteq L$ is a* telling set *for L if $(D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(S) \cap D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L)) \neq \emptyset$.*

Note that telling sets have some similarity to the concept of telltales that is used in the model of learning in the limit. For a comparison of telltales and telling sets, see our comments after Corollary 32.

Using Lemma 14, we are now able to show that the existence of a telling set is characteristic for the correct convergence of Canon on any text:

Theorem 25 *Let Σ an alphabet with $|\Sigma| \geq 2$. For every language $L \subseteq \Sigma^*$, and every text $t \in \mathrm{text}(L)$,* Canon *converges correctly on t if and only if L has a telling set.*

In the final part of this section, we shall demonstrate that this is a strong result, by investigating the existence and nonexistence of telling sets for various languages.

4.3 Examination of the Class $\mathcal{TSL}_\Sigma$

As stated by Theorem 25, the existence of telling sets is a strong sufficient criterion for $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$-descriptive generalisability. Furthermore, generalisability of a class $\mathcal{L} \subseteq \mathcal{P}(\Sigma^*)$ using Canon does not depend on the properties of the whole class, but only on the existence of a telling set for every single language $L \in \mathcal{L}$. Thus, we consider the largest possible class that can be generalised by Canon and define $\mathcal{TSL}_\Sigma := \{L \subseteq \Sigma^* \mid L \text{ has a telling set}\}$. Theorem 25 immediately leads to the following corollary:

Corollary 26 *For every alphabet Σ with $|\Sigma| \geq 2$, $\mathcal{TSL}_\Sigma \in \mathrm{DG}_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}$.*

Thus, by examining $\mathcal{TSL}_\Sigma$, we gain insights into the power of Canon and of the whole model of descriptive generalisation. Before we proceed to an examination of the relation of various classes of languages to $\mathcal{TSL}_\Sigma$, we show that it is not required to choose Σ as small as possible, a result that is similar to Corollary 16, which states that $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L)$ is largely independent of the choice of Σ. The same holds for telling sets:

Corollary 27 *Let Σ, Σ' be alphabets with $|\Sigma|, |\Sigma'| \geq 2$. Then $L \in \mathcal{TSL}_\Sigma$ if and only if $L \in \mathcal{TSL}_{\Sigma'}$ for every $L \subseteq (\Sigma \cap \Sigma')$.*

This also implies that, for every $\Sigma' \supseteq \Sigma$, $\mathcal{TSL}_{\Sigma'} \supseteq \mathcal{TSL}_\Sigma$. We begin our examination of $\mathcal{TSL}_\Sigma$ by expanding finite languages without losing their telling set properties. The next result follows immediately from Lemmas 14 and 17:

Lemma 28 *Let Σ be an alphabet with $|\Sigma| \geq 2$. Every nonempty $S \in \mathrm{FIN}_\Sigma$ is a telling set of* $\text{S-Hull}_\Sigma(S)$ *and of every L with $S \subseteq L \subseteq$* $\text{S-Hull}_\Sigma(S)$.

In addition to showing that $\mathrm{FIN}_\Sigma \subseteq \mathcal{TSL}_\Sigma$, this result allows us (in conjunction with Lemma 22) to make the following statement on the cardinality of $\mathcal{TSL}_\Sigma$:

Proposition 29 *$\mathcal{TSL}_\Sigma$ is uncountable for every alphabet Σ with $|\Sigma| \geq 2$.*

This is an uncommon property, as inference from positive data is normally considered for classes consisting of countably many languages from some countable domain. Nonetheless, inferrability of uncountable classes has been studied before, see Jain et al. (2009).

Next, we shall see that $\mathcal{TSL}_\Sigma$ contains a rich and natural class of languages, the DTF0L languages. A DTF0L language L over Σ is defined through a finite set of axioms $w_1, \ldots, w_m \in \Sigma^*$ and a finite set of morphisms $\phi_1, \ldots, \phi_n : \Sigma^* \to \Sigma^*$. Then L is the smallest language that satisfies $w_i \in L$ for every $i \in \{1, \ldots, m\}$, and if $w \in L$, then $\phi_i(w) \in L$ for every $i \in \{1, \ldots, n\}$. We denote the class of all DTF0L languages over Σ by $\mathrm{DTF0L}_\Sigma$. Apart from FIN_Σ, the most prominent subclass of $\mathrm{DTF0L}_\Sigma$ is the class of D0L languages, where every language is defined through a single axiom and a single morphism (i. e., $m = n = 1$). The class D0L has been widely studied, for details, see Kari et al. (1997).

Proposition 30 *Let Σ be an alphabet with $|\Sigma| \geq 2$. Then $\mathrm{DTF0L}_\Sigma \subseteq \mathcal{TSL}_\Sigma$.*

Lemma 28 and Proposition 30 both imply that $\mathrm{FIN}_\Sigma \subseteq \mathcal{TSL}_\Sigma$. Furthermore, Proposition 29 and Proposition 30 both demonstrate that $\mathcal{TSL}_\Sigma$ contains at least one infinite language, which leads to the following observation:

Corollary 31 *The class $\mathcal{TSL}_\Sigma$ is superfinite for every alphabet Σ with $|\Sigma| \geq 2$.*

Together with Proposition 23, this allows us to describe the relation between $\mathrm{DG}_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}$ and LIM-TEXT:

Corollary 32 *Let Σ be an alphabet, $|\Sigma| \geq 2$. Then $\mathrm{DG}_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}$ and LIM-TEXT are incomparable.*

We now briefly discuss the relation between telling sets and the notion of telltales. As already mentioned in Section 3.2, according to Angluin (1980b), an indexed family $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$ of non-empty recursive languages is in LIM-TEXT if and only if there exists an effective procedure which, for every $j \geq 0$, enumerates a set T_j such that T_j is finite, $T_j \subseteq L_j$, and there does not exist a $j' \geq 0$ with $T_j \supseteq L_{j'} \supset L_j$. If there exists a set T_j satisfying these conditions, it is called a *telltale* for L_j with respect to $\mathcal{L} = (L_i)_{i \in \mathbb{N}}$. Thus, the concepts of telltales and telling sets are incomparable, as the former refers to a language and the class of languages it is contained in, whereas the latter relates to a language and certain properties of the class $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$. Nevertheless, for every language L in $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$, a set S is a telling set for L if and only if S is a telltale for L with respect to $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$ (for more details on the existence of telltales for languages in $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$, see Reidenbach (2008)).

As Proposition 33 and Proposition 34 below show, Lemma 25 by Reidenbach (2008) and Lemma 7 by Reidenbach (2006) on the existence and nonexistence of telltales lead to the corresponding results for telling sets:

Proposition 33 *Let Σ, Σ' be alphabets, $\Sigma' \subseteq \Sigma$ and $|\Sigma'| \geq 3$. For every $\alpha \in X^+$, $L_{\mathrm{E},\Sigma'}(\alpha)$ has a telling set.*

On the other hand, it is impossible to encode the structure of comparatively simple patterns in their languages with only two letters, which leads to the following negative result:

Proposition 34 *Let Σ be an alphabet with $|\Sigma| \geq 2$, and let $\mathtt{a}$, $\mathtt{b}$ be two distinct letters from Σ. Then $L_{\mathrm{E},\{\mathtt{a},\mathtt{b}\}}(x_1^2 x_2^2 x_3^2) \notin \mathcal{TSL}_\Sigma$.*

In contrast to this, Lemma 20 can be used to show that restricting the number of variables in the patterns leads to telling sets not only for languages from $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$, but also for their finite unions:

Proposition 35 *Let $\alpha_1, \ldots, \alpha_n \in \{x_1, \ldots, x_{|\Sigma|}\}^+$, and let $L := \bigcup_{i=1}^n L_{\mathrm{E},\Sigma}(\alpha_i)$. Then $L \in \mathcal{TSL}_\Sigma$.*

Proposition 35 is especially interesting when compared to Proposition 21, which tells us that infinite unions of languages from $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$ might not only have no telling set, but not even a descriptive pattern.

Furthermore, we state that the infinite sequence $(\beta_n)_{n \geq 0}$ that is used in the definition of the languages L_Σ for the proof of Theorem 4 describes an infinite ascending chain of languages from $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$; i. e., $L_{\mathrm{E},\Sigma}(\beta) \subset L_{\mathrm{E},\Sigma}(\beta_{n+1})$ for every $n \geq 0$. Although the presence of such a chain in $\text{S-Hull}_\Sigma(L)$ for a language L does not necessarily imply emptiness of $D_{\mathrm{ePAT}_{\mathrm{tf},\Sigma}}(L)$, it is a sufficient criterion for $L \notin \mathcal{TSL}_\Sigma$ (again, the proof relies on Lemma 17):

Lemma 36 *Let Σ be an alphabet with $|\Sigma| \geq 2$ and let $L \subseteq \Sigma^*$. If there is an infinite chain $(\beta_n)_{n \geq 0}$ over X^+ with $L_{\mathrm{E},\Sigma}(\beta_n) \subseteq \text{S-Hull}_\Sigma(L)$ for every $n \geq 0$, $L_{\mathrm{E},\Sigma}(\beta_n) \subset L_{\mathrm{E},\Sigma}(\beta_{n+1})$ for every $n \geq 0$, and $\bigcup_{n \geq 0} L_{\mathrm{E},\Sigma}(\beta_n) \supseteq L$, then L has no telling set.*

As a direct application of this result, we can prove that there are regular languages that have no telling set:

Proposition 37 *For every alphabet Σ with $|\Sigma| \geq 2$, there is a regular language $L \subseteq \Sigma^*$ with $L \notin \mathcal{TSL}_\Sigma$.*

Note that this language is also an example of a language L that has no telling set, although $\text{S-Hull}_\Sigma(L)$ has a telling set.

Acknowledgements

The authors wish to thank Steffen Lange for his helpful remarks and suggestions on the inference model and some results presented in this paper.

References

Angluin, D. (1980a). Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, *21*, 46–62.

Angluin, D. (1980b). Inductive inference of formal languages from positive data. *Information and Control*, *45*, 117–135.

Arimura, H., Shinohara, T., & Otsuki, S. (1994). Finding minimal generalizations for unions of pattern languages and its application to inductive inference from positive data. *Proc. STACS 1994* (pp. 649–660).

Câmpeanu, C., Salomaa, K., & Yu, S. (2003). A formal study of practical regular expressions. *International Journal of Foundations of Computer Science*, *14*, 1007–1018.

Ehrenfeucht, A., & Rozenberg, G. (1979). Finding a homomorphism between two words is NP-complete. *Information Processing Letters*, *9*, 86–88.

Freydenberger, D., & Reidenbach, D. (2009). Existence and nonexistence of descriptive patterns. *Proc. DLT 2009* (pp. 228–239).

Freydenberger, D., & Reidenbach, D. (2010). Bad news on decision problems for patterns. *Information and Computation*, *208*, 83–96.

Fulk, M., & Jain, S. (1996). Learning in the presence of inaccurate information. *Theoretical Computer Science*, *161*, 235–261.

Gold, E. (1967). Language identification in the limit. *Information and Control*, *10*, 447–474.

Holub, S. (2009). Polynomial-time algorithm for fixed points of nontrivial morphisms. *Discrete Mathematics*, *309*, 5069–5076.

Jain, S., & Kinber, E. (2008). Learning and extending sublanguages. *Theoretical Computer Science*, *397*, 233–246.

Jain, S., Luo, Q., Semukhin, P., & Stephan, F. (2009). Uncountable automatic classes and learning. *Proc. ALT 2009* (pp. 293–307).

Jiang, T., Kinber, E., Salomaa, A., Salomaa, K., & Yu, S. (1994). Pattern languages with and without erasing. *International Journal of Computer Mathematics*, *50*, 147–163.

Jiang, T., Salomaa, A., Salomaa, K., & Yu, S. (1995). Decision problems for patterns. *Journal of Computer and System Sciences*, *50*, 53–63.

Kari, L., Rozenberg, G., & Salomaa, A. (1997). L systems. In G. Rozenberg and A. Salomaa (Eds.), *Handbook of Formal Languages*, vol. 1, chapter 5, 253–328. Springer.

Kobayashi, S., & Yokomori, T. (1995). On approximately identifying concept classes in the limit. *Proc. ALT 1995* (pp. 298–312).

Kobayashi, S., & Yokomori, T. (1997). Learning approximately regular languages with reversible languages. *Theoretical Computer Science*, *174*, 251–257.

Lange, S., Zeugmann, T., & Zilles, S. (2008). Learning indexed families of recursive languages from positive data: A survey. *Theoretical Computer Science*, *397*, 194–232.

Mukouchi, Y. (1994). Inductive inference of an approximate concept from positive data. *Proc. ALT 1994* (pp. 484–499).

Reidenbach, D. (2006). A non-learnable class of E-pattern languages. *Theoretical Computer Science*, *350*, 91–102.

Reidenbach, D. (2008). Discontinuities in pattern inference. *Theoretical Computer Science*, *397*, 166–193.

Reidenbach, D., & Schneider, J. (2009). Morphically primitive words. *Theoretical Computer Science*, *410*, 2148–2161.

Rogers, H. (1992). *Theory of recursive functions and effective computability*. Cambridge, MA: MIT Press. 3rd print.

Rozenberg, G., & Salomaa, A. (1997). *Handbook of Formal Languages*, vol. 1. Berlin: Springer.